

Project-Bragboard

Name: Lakshmi Prasanna

milestone-1:Task:2

overview:

In this milestone, the focus is on establishing the foundational structure of the application both in the frontend and backend. The goal is to create a visually clean, responsive interface and implement initial backend logic to support user-specific and department-specific data access.

- 1. Basic Layout with Tailwind CSS:**
The frontend is designed using Tailwind CSS to ensure a modern and responsive layout. This includes setting up the overall structure of the dashboard, navigation bar, and content sections. Tailwind's utility-first classes help in rapid UI development and consistent styling across all components.
- 2. User Dashboard Display After Login:**
Once a user successfully logs in, they are redirected to a personalized dashboard. The dashboard serves as the central interface where users can view relevant data, notifications, and functionalities based on their role or department. Session management ensures only authenticated users can access this section.
- 3. Department-wise Scoping in Backend:**
On the backend, department-based data access is configured to ensure each user can only interact with resources related to their specific department. This involves implementing role-based authorization logic and defining API endpoints that return filtered data according to the user's department scope.

Errors:**Missing table column error** – occurs when a new column isn't added in the database migration.

Tailwind not applying styles – happens if the config or import is incorrect.

Project Setup and Environment Configuration

- A monorepo structure was created, separating the project into a dedicated backend directory for the FastAPI application and a frontend directory for the React.js application.
- The backend environment was configured with a Python virtual environment and all necessary dependencies, including SQLAlchemy for database interactions.
- The frontend environment was initialized using Vite, and Tailwind CSS was installed and configured for efficient styling.
- A scalable folder structure was implemented for both backend and frontend to ensure maintainability:

- Backend: models, schemas, crud, api
- Frontend: pages, components, services

Instructions for Running the Project

To run the BragBoard project, both the backend (FastAPI) and frontend (React) need to be set up and running. Follow these steps:

Backend (FastAPI)

1. Open a terminal and navigate to the backend directory: `cd backend`
2. Create and activate a Python virtual environment:
 - Windows: `python -m venv venv → venv\Scripts\activate`
 - macOS/Linux: `python -m venv venv → source venv/bin/activate`
3. Install the required dependencies using `pip install -r requirements.txt`
4. Start the FastAPI server using: `uvicorn app.main:app --reload`
 - The backend will run at <http://127.0.0.1:8000/docs>

Frontend (React with Vite)

1. Open a terminal and navigate to the frontend directory: `cd frontend`
2. Install the required packages using `npm install`
3. Start the development server with `npm run dev`
 - The frontend will run at <http://localhost:5173>

Project Repository

The source code for BragBoard is hosted on GitHub and can be accessed at the following link:

[BragBoard Repository](#).